Short-term Electricity Load Forecasting Based on Particle Swarm Algorithm and SVM

Liping Wang   Jingmin Wang   Dan Zhao

Economics and Management Department, North China Electric Power University, Baoding 071003, P.R.China

Abstract

In electricity industry, accurate load forecasting plays a key role in assuring the stability of power network and society. By far, there are many methods and models proposed to enhance the accuracy of forecasting results. On the basis of analyzing the performance of particle swarm algorithm (PSA) and SVM, the paper proposed a new forecasting model which is proved to be able to enhance the accuracy, improve the convergence ability and reduce operation time by numerical experiment. The proposed model is expected to offer a valid alternative for application in the load forecasting field.

Keywords: Particle swarm algorithm, Support vector machines, Load forecasting

1. Introduction

With the rapid development of electricity industry, accurate forecasting of short-term electricity load has received growing attention. Hence, during recent decades, many researchers in load forecasting field tried their best to study load forecasting techniques and models.

One such method is a weather insensitive approach that uses historical load data to forecast future electricity load. Generally, it is known as the Box–Jenkins autoregressive integrated moving average (ARIMA) [1-3]. Christianse [4] and Park et al. [5] designed exponential smoothing models by Fourier series transformation for electricity load forecasting. Mbamalu and El-Hawary [6] proposed multiplicative auto-regressive(AR) models that considered seasonal factors in load forecasting. The analytical results showed that the forecasting accuracy of the proposed models outperformed the univariate AR model. Douglas et al. [7] considered verifying the impacts of temperature on the forecasting model. The authors combined Bayesian estimation with a dynamic linear model for load forecasting. The experimental results demonstrated that the presented model is suitable for forecasting load under imperfect weather information. Sadownik and Barbosa [8] proposed dynamic nonlinear models for load forecasting. The main disadvantage of these methods is that they become time consuming to compute as the number of variables increases. To achieve accurate load forecasting, the load forecasting model employed state space and Kalman filtering technology, which were developed to reduce the difference between the actual and predicted loads. Moghram and Rahman [9] devised a model based on the state space and Kalman filtering technology, and verified that the proposed model outperformed four other forecasting methods. The disadvantage of these methods is the difficulty of avoiding observation noise in the forecasting.

The regression approach is another popular model for forecasting electricity load. Regression models construct the cause–effect relationships between electricity load and the independent variables. The most popular models are linear regression, proposed by Asbury[10], which considers the ‘‘weather’’ variable in the forecasting model. Meanwhile, Papalexopoulos and Hesterberg [11] added ‘‘holiday’’ and ‘‘temperature’’ to the model. Furthermore, Solimanetal. [12] designed a multivariate linear regression model in load forecasting. These models were assumed to be linear and are computationally intensive. However, these independent variables were not justified for use because the terms are known to be nonlinear.

With the development of artificial intelligence, the artificial neural network was introduced into load forecasting field to enhance the forecasting performance. Rahman and Bhatnagar [13] presented a knowledge based expert system (KBES) approach for electricity load forecasting. Park et al. [14] established a three layer back propagation neural network to implement daily load forecasting problems. Moreover, Hoetal. [15] developed an adaptive learning algorithm for forecasting the electricity load in Taiwan. Novak [16] applied radial basis function (RBF) neural networks to forecast electricity load. However, these models cannot solve the same problems of minimizing the expected risk and determining network structure scientifically.

Recently, support vector machine (SVM) proposed by Vapnik, which achieves the structure risk and experience risk minimization, and also minimizes the boundary of vapnik chervonenks(VC) dimension, was applied to electricity load forecasting. Cao [17] used the SVMs experts for time series forecasting. The generalized SVMs experts contained a two stage neural network architecture. The numerical results indicated that the SVMs experts are capable of outperforming the single SVMs models in terms of generalization comparison. Cao and Gu [18] proposed a dynamic SVMs model to deal with non-stationary time series problems. Experimental results showed that the dynamic SVMs model outperform standard SVMs in forecasting non-stationary time series. Meanwhile, Tay andCao [19] presented C-ascending SVMs to model non-stationary financial time series. Experimental results showed that the C-ascending SVMs with actually ordered sample data consistently perform better than standard SVMs. Tay and Cao [20] used SVMs in forecasting financial time series. The numerical results indicated that theSVMs are superior to the multi-layer back propagation neural network in financial time series forecasting. Lu et al.[21]applied SVMs in predicting air quality parameters with different time series.The experimental results indicated that SVMs outperform the conventional radial basis function (RBF)networks. However, how to reduce input parameters of SVM and select key influencing factors is still a problem for SVM-based load forecasting.

This paper attempts to introduce particle swarm algorithm into SVM-based load forecasting to overcome the shortcomings of above load forecasting methods. The experimental results reveal that proposed model outperforms SVMSA model proposed by other researchers.

2. New model

2.1. Particle swarm algorithm

Particle swarm algorithm (PSA), as a kind of global search algorithm, was proposed by Kennedy and Eberhart in 1995[22]. It searches for the optimal value by sharing historical information and social information between the particle individuals. Many examples have showed that this algorithm has many advantages, such as simple concept and good convergence capacity, and can be applied in some optimization fields. PSA is based on swarm behavior, which can move the individuals to the best positions according to their fitness to the environment. Every individual in the swarm is regarded as a particle that has no volume and can fly in a D-dimensional search space at a fixed velocity, which is regulated by the flight experience of the particle itself and those of other particles.

Suppose that m particles form themselves into a swarm in a D-dimensional search space. X_i represents the position of the ith particle, which is marked as $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$; V_i is the velocity of the ith particle, which is marked as $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$; the best position of some particle is $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the best position of the swarm is $p_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. Hence, the position and velocity of the particles can be expressed with the Eq.1 and Eq.2

$$\{v_{id}(k+1) \\ = \omega \cdot v_{id}(k) + rand(0, c_1) \cdot [p_{id}(k) - x_{id}(k)] \\ + rand(0, c_2) \cdot [p_{gd}(k) - x_{id}(k)] \qquad (1)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1), d = 1,2,\cdots, D \qquad (2)$$

where ω is an inertial weight which represents the influence of the previous velocity of a particle upon its current velocity. The bigger the ω, the bigger the velocity v_{id}, the bigger the search space for the particles, which helps find new solution spaces. The smaller the ω, the smaller the v_{id}, which helps find a better solution in the current space. c_1 and c_2 are acceleration constants. $rand(0, c_1)$ and $rand(0, c_2)$ are the random numbers evenly distributed, respectively, in $[0, c_1]$ and $[0, c_2]$. If c_1=0, then the particle only has 'self-experience', its convergence rate may be fast, and it is easy to fall into the local optimum. If c_2 =0, then the particle only has 'social experience', all particles become moving by themselves without interaction, and the probability of finding a solution is little. If $c_1 = c_2 = 0$, then the particle has no any 'experience', and all particles become disorderly and unsystematic. Generally, v_{id} is conditioned by $v_{id} \in [-v_{\max}, v_{\max}]$ to stop the particles from flying out of the solution area.

The basic steps for PSA:

Step1: Initializing all particles. Set the initial position p_i and the initial velocity v_{id} of all particles randomly, and optimal initial position is regarded as the best position p_g of swarm;

Step2: Calculating the fitness value of all particles;

Step3: Comparing the current fitness value of each particle with its own historical best position p_i, if its own historical best position p_i is smaller, then it is replaced with the current fitness;

Step4: Comparing the best current position of all particles with the historical best position p_g of the swarm, if p_g is smaller, then it is replaced with the best current position of all particles;

Step5: Refreshing the positions and velocities of all particles according to the Eq.1 and Eq.2;

Step6: Judging whether the termination criterion is satisfied, if 'Yes', then stopping the iteration operation; if 'No', then going back to Step 2. The termination criterion could be a given iteration steps, or a given fitness of the particles, or the standstill of the optimal solution.

2.2. SVM regression

Following the introduction of the ε insensitive loss function, SVMs have been used to solve Non-linear regression problems. The basic concept of the SVM regression is to map nonlinearly the original data x into a higher dimensional feature space. Hence, given a set of data $G=\{(x_i,a_i)\}_{i=1}^N$ (where x_i is the input vector; a_i is the actual value and N is the total number of data patterns), the SVM regression function is

$$y=f(x)=\omega_i\varphi_i(x)+b \qquad (3)$$

Where $\varphi_i(x)$ is called the feature that is nonlinearly mapped from the input space x. The ω_i and b are coefficients that are estimated by minimizing the regularized risk function

$$R(C)=C\frac{1}{N}\sum_{i=1}^{N}\Theta_\varepsilon(d_i,y_i)+\frac{1}{2}\|\omega\|^2 \qquad (4)$$

where

$$\Theta_\varepsilon(a,y)=\begin{cases}0 & |d-y|\le\varepsilon\\ |d-y|-\varepsilon & otherwise\end{cases} \qquad (5)$$

and C and ε are prescribed parameters. In Eq. (4), $\Theta_\varepsilon(d,y)$ is called the ε insensitive loss function. The loss equals zero if the forecasted value is within the ε tube (Eq. (5) and Fig. 1). The second term, $\frac{1}{2}\|\omega\|^2$, measures the flatness of the function. Therefore, C is considered to specify the trade off between the empirical risk and the model flatness. Both C and ε are user determined parameters. Two positive slack variables ξ and ξ^*, which represent the distance from the actual values to the corresponding boundary values of ε tube (Fig. 1), are introduced. Then, Eq. (4) is transformed into the following constrained form: Minimize

$$R(\omega,\xi,\ \xi^*)=\frac{1}{2}\|\omega\|^2+C(\sum_{i=1}^{N}(\xi_i+\xi_i^*)) \qquad (6)$$

With the constraints,

$$\omega_i\varphi(x_i)+b_i-d_i\le\varepsilon+\xi_i^*,\ i=1,2,\cdots,N \qquad (7)$$

$$d_i-\omega_i\varphi(x_i)-b_i\le\varepsilon+\xi_i,\ i=1,2,\cdots,N \qquad (8)$$

$$\xi_i,\xi_i^*\ge 0, i=1,2,\cdots,N$$

This constrained optimization problem is solved using the following primal Lagrangian form:

$$L(\omega_i,b,\xi,\xi^*,\alpha_i,\alpha_i^*,\beta_i,\beta_i^*)=\frac{1}{2}\|\omega\|^2+C(\sum_{i=1}^{N}(\xi_i+\xi_i^*))-\sum_{i=1}^{N}\alpha_i[\omega_i\varphi(x_i)+b-d_i+\varepsilon+\xi_i]$$

$$-\sum_{i=1}^{N}\alpha_i^*[d_i-\omega_i\varphi(x_i)-b+\varepsilon+\xi_i^*]-\sum_{i=1}^{N}(\beta_i\xi_i+\beta_i^*\xi_i^*) \qquad (9)$$

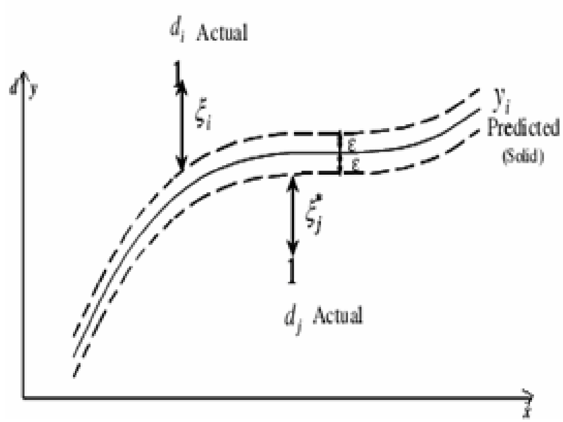

Fig.1: Parameters used in SVM regression.

Eq. (9) is minimized with respect to the primal variable ω_i, b, ξ and ξ^*, and maximized with respect to the non-negative Lagrangian multipliers α_i, α_i^*, β_i and β_i^*. Finally, the Karush-Kuhn-Tucker conditions, which is applied to obtain an optimal solution of Eq. (9), are applied to the regression, and Eq. (6), thus, yields the dual Lagrangian,

Subject to the constrainsts,

$$\vartheta(\alpha_i,\alpha_i^*)=\sum_{i=1}^{N}\alpha_i(\alpha_i-\alpha_i^*)-\varepsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j) \qquad (10)$$

$$\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0\quad,\quad 0\le\alpha_i\le C, i=1,2,\cdots,N$$

$$0\le\alpha_i^*\le C, i=1,2,\cdots,N$$

The Lagrange multipliers in Eq. (10) satisfy the equality $\alpha_i^* \alpha_i^* = 0$. The Lagrange multipliers and α_i^* are calculated and an optimal desired weight vector of the regression hyperplane is

$$\omega^* = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(x, x_i) \quad (11)$$

Hence, the regression function is

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (12)$$

Where $K(x, x_i)$ is called the kernel function. The value of the kernel is equal to the inner product of the two vectors x and x_i in the feature space $\varphi(x)$ and $\varphi(x_i)$, i.e. $K(x, x_i) = \varphi(x) * \varphi(x_i)$. Any function that satisfies Mercer's condition can be used as the kernel function. The matrix K, $K = (K(x, x_j))_{j=1}^{N}$, satisfies Mercer's condition when all its eigenvalues are greater than or equal to zero. In this work, the Gaussian function, $\exp(-\frac{1}{2} * (\frac{\| x_i - x_j \|}{\sigma})^2)$, is used in the SVM.

2.3. Optimization algorithm of input variables based on PSA

For an input variables set $F = \{f_1, f_2, \cdots, f_N\}$, where N is the size of the variables set, f_i represents an input variable. The following binary vector is introduced to stand for the feature selection: $S = \{s_1, s_2, \cdots, s_N\}$, $s_i \in [0,1]$, $i = 1, 2, \cdots, N$. The value 0 and 1 for s_i respectively, stand for whether the f_i in set is selected or not. The SVM performance E is regarded as the target function. Hence, the Optimization of input variables selection of SVM can be expressed as

$$\max_{S} E(S)$$

$$\text{s.t.} \quad \begin{aligned} & S = \{s_1, s_2, \cdots, s_N\}, \\ & s_i \in [0,1], i = 1, 2, \cdots, N \end{aligned} \quad (13)$$

This optimization problem can be solved by PSA, in which the variable selection S serves as the individual set whose fitness is the SVM performance E.

The basic steps as follows:

Step1: Initializing all particles. The parameters such as σ, ε and C should be involved in every particle. The parameters are encoded in this way: In PSA, the swarm has m particles, each particle is a 3-dimensional vector that contains the SVM parameters such as σ, ε and C. The coding of ith particle is defined as $X_i = (x_{i1}, x_{i2}, x_{i3})$, where x_{i1}, x_{i2}, x_{i3} respectively denotes σ, ε and C. Each parameter is randomly valued in its limited area at the Initialisation of the particle swarm.

Step2: Calculating the fitness value of all particles;

Step3: Comparing the current fitness value of each particle with its own historical best position p_i, if its own historical best position p_i is smaller, then it is replaced with the current fitness;

Step4: Comparing the best current position of all particles with the historical best position p_g of the swarm, if p_g is smaller, then it is replaced with the best current position of all particles;

Step5: Refreshing the positions and velocities of all particles according to the Eq.1 and Eq.2;

Step6: Judging whether the termination criterion is satisfied, if 'Yes', then stopping the iteration operation; if 'No', then going back to Step 2. The termination criterion could be a given number of the iteration times, or an operating time for the computer, or a data accuracy attained by the iteration. In this paper, the termination criterion stands for iteration times that is 60, and that the fitness must be obviously raised in any 20 successive iterative steps.

If the preset iteration times is reached, or the fitness is not obviously raised in any 20 successive iterative steps, the termination criterion is considered satisfied, and the iteration procedure halts.

2.4. Forecasting using new model

Step 1: optimizing and selecting input variables of SVM based on algorithm in Section 2.3

According to the characteristic of load change in a day, we select the temperature information of forecasted day、the past three days and the corresponding days of past two weeks as temperature genes, marked as $\{dt, d(t-1), d(t-2), d(t-3), d(t-7), d(t-14)\}$, and the historical load data of the corresponding time point and 3 hours before that of past three days, the corresponding time point and 3 hours before that of the corresponding day of past two weeks, marked as $L(i, j)$, where d and t respectively represent forecasted day and forecasting time, $i = \{d-1, d-2, d-3, d-7, d-14\}$, $j = \{t-3, t-2, t-1, t\}$.

Therefore, the total number of input variables of SVM is 68.

Because the data processed by PSA should be discrete, we should disperse the sequential data before optimization. There are many methods to disperse sequential data. In this paper, we adopt the dispersing minimal information entropy algorithm.

After this process, we can mine the effective data to get the optimized input variables set, which is marked as $L(d-1,t)$ 、 $L(d-1,t-1)$ 、 $L(d-2,t)$ 、 $L(d-2,t-1)$ 、 $L(d-7,t)$ 、 $T_{\min,d}$ 、 $T_{\text{avg},d}$ 、 $T_{\text{avg},d-1}$ 、 H_d and H_{d-1}, where $L(d,t)$ is load of time point t of forecasted day, $T_{\min,d}$ is the minimal temperature of forecasted day, $T_{\text{avg},d}$ is the average temperature of forecasted day, and H_d is the humidity of forecasted day.

Step 2: SVM and load forecasting

According to the type of day load, we classify the load into two classes, work time load and weekend load. 24 SVM models are built for every kind of load to forecast the load value of 24 hours in forecasted day. The optimized input variables got in Step1 is used as final input variables of SVM, and LIBSVM is adopted as training software. The training results of SVM are used to forecast the short-term electricity load.

3. Numerical example

This example used load data of some area in Shanghai from 7, 1, 2004 to 9, 1, 2004, which can gotten from Reference 23, to forecast the day load from 8, 1, 2004 to 8, 10, 2004 so as to demonstrate the forecasting effect and compare the forecasting performance of the new model and SVMSA model proposed in Reference 24.

The mean absolute percent error (MAPE) of forecasting result is

$$E = \frac{1}{N}\sum_{i=1}^{N}\frac{|R(i)-F(i)|}{R(i)} \qquad (14)$$

Where $R(i)$ and $F(i)$ respectively represent the actual value and forecasted value of load, and N=24.

The MAPE values of forecasting results by new model and SVMSA model are showed in Table 1.

It is obvious that the new model presented has higher accuracy than the SVMSA model. Moreover, the forecasting processes of these two models reveal that the new model has better convergence ability and consume less time than the other two models.

To demonstrate accuracy and good convergence ability of new model, we respectively adopt these two models to forecast load values of 24 time points on 08, 08, 2004. The forecasting results and relative error are showed in Table 2. The load curves gotten by these two models are plotted in Fig. 2.

4. Conclusions

In electricity industry, accurate load forecasting plays a key role in assuring the stability of power network and society. On the basis of analyzing the

Day	New Model	SVMSA Model
	MAPE/%	MAPE/%
08,01,2004	1.8318	2.0652
08,02,2004	1.2451	1.3843
08,03,2004	1.3648	1.6528
08,04,2004	1.0653	1.0549
08,05,2004	1.9828	2.0682
08,06,2004	2.1874	2.6619
08,07,2004	1.2845	1.9533
08,08,2004	1.0998	1.2562
08,09,2004	1.5284	1.9968
08,10,2004	1.3524	1.5624
Average value	1.4942	1.7656

Table 1: Comparison of load forecasting results.

Time Point	New Model		SVMSA Model		Actual Load /MW
	Forecasted result/MW	Relative Error/%	Forecasted result/MW	Relative Error/%	
00:00	557.1985	0.3563	558.3842	0.5699	555.22
01:00	508.4877	0.6468	510.6525	1.0753	505.22
02:00	462.2445	-0.393	471.6853	1.641	464.07
03:00	438.2548	0.5587	437.9685	0.493	435.82
04:00	417.8325	0.5686	418.5560	0.7428	415.47
05:00	403.6997	0.5554	398.8474	-0.653	401.47
06:00	409.8234	-0.752	415.8895	0.7167	412.93
07:00	448.3784	0.5761	463.2896	3.9209	445.81
08:00	535.6455	1.4865	538.6651	2.0586	527.80
09:00	638.9413	0.8223	640.8553	1.1243	633.73
10:00	716.5421	0.1582	713.5426	-0.261	715.41
11:00	768.4993	0.4154	772.3315	0.9162	765.32
12:00	808.7852	0.8624	810.6524	1.0952	801.87
13:00	835.4866	0.313	840.6645	0.9346	832.88
14:00	826.9983	0.4895	829.7844	0.828	822.97
15:00	814.8782	0.5749	815.9986	0.7132	810.22
16:00	818.6651	0.9813	816.4579	0.709	810.71
17:00	808.1154	0.6859	809.8775	0.9055	802.61
18:00	794.9282	0.8818	796.5237	1.0843	787.98
19:00	817.7558	-0.178	810.3526	-1.081	819.21
20:00	826.6425	1.3477	828.9748	1.6336	815.65
21:00	803.3597	1.5022	809.5586	2.2854	791.47
22:00	704.2674	0.6686	695.6298	-0.566	699.59
23:00	627.8865	1.6146	628.5536	1.7225	617.91
MAPE	0.7246%		1.1555%		

Table 2: Comparison of load forecasting result.

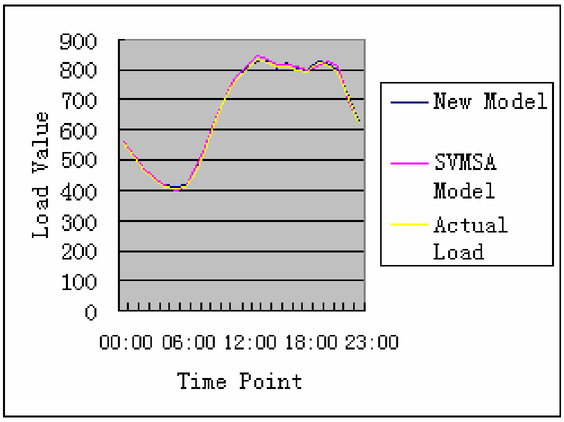

Fig.2: The load curves.

performance of particle swarm algorithm (PSA) and SVM, the paper proposed a new forecasting model which is proved to be able to enhance the accuracy, improve the convergence ability and reduce operation time by numerical experiment. The results of numerical experiment also indicate that the new model outperformed the other model proposed by some researchers. There are many causes for new model to have so superior performance. Firstly, PSA can reduce the input variables of SVM. Secondly, this new model can select the effective variables in a shorter time, improve the performance of the SVM classifier, and has fewer errors and a better real-time capacity than the SVMSA model.

This work is the first to apply the SVM model with PSA to short-term electricity load forecasting. The experimental results showed that the proposed model can offer a valid alternative for application in the load forecasting field.

References

[1] B. Gep and J. Gm, *Time series analysis, Forecasting and Control*, San Francisco, Holden Day, 1970.

[2] S. Vemuri, D. Hill and R.Balasubramanian, Load forecasting using stochastic models. *Proceeding of the 8th Power Industrial Computing Application Conference*, 1:31–37, 1973.

[3] J.F. Chen, W.M. Wang and C.M. Huang, Analysis of an adaptive time-series autoregressive moving-average(ARMA) model for short-term load forecasting. *Electric Power Syst*, 34: 187–196, 1995.

[4] W.R. Christianse, Short term load forecasting using general exponential smoothing. *IEEE Trans Power Apparatus Syst*, 90: 900–911, 1971.

[5] J.H. Park, Y.M. Park and K.Y. Lee, Composite modeling for adaptive short-term load forecasting. *IEEE Trans Power Syst*, 6: 450–457, 1991.

[6] M. Gan and E. Me, Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation. *IEEE Trans Power Syst*, 8:343–348, 1993.

[7] A.P. Douglas, A.M. Breipohl, FN. Lee and R. Adapa, The impact of temperature forecast uncertainty on Bayesian load forecasting. *IEEE Trans Power Syst*, 13: 507–1513, 1998.

[8] R. Sadownik and E.P. Barbosa, Short-term forecasting of industrial electricity consumption in Brazil. J Forecast, 18: 215–224, 1999.

[9] I. Moghram and S. Rahman, Analysis and evaluation of five short-term load forecasting techniques. *IEEE Trans Power Syst*, 4: 1484–1491, 1989.

[10] C. Asbury, Weather load model for electric demand energy forecasting. *IEEE Trans Power Apparatus Syst*, 94: 1111–1116, 1975.

[11] A.D. Papalexopoulos and TC. Hesterberg, Aregression based approach to short-term systerm load forecasting, *IEEE Trans Power Syst*, 5:1535–1547, 1990.

[12] S.A. Soliman, S. Persaud, K. El-Nagar, ME. El-Hawary, Application of least absolute value parameter estimation based on linear programming to short-term load forecasting. *Electrical Power Energy Syst*, 19: 209–216, 1997.

[13] S. Rahman and R. Bhatnagar, An expert system based algorithm for short-term load forecasting. *IEEE Trans Power Syst*, 3: 292–299, 1988.

[14] D.C. Park, M.A. El-Sharkawi, RJ. Marks II, LE.Atlas and MJ.Damborg, Electric load forecasting using an artificial neural network. *IEEE Trans Power Syst*, 6: 442–449, 1991.

[15] K.L. Ho, Y.Y. Hsu and C.C. Yang, Short-term load forecasting using a multilayer neural network with an adaptive learning algorithm. *IEEE Trans Power Syst*, 7: 141–149, 1992.

[16] B. Novak, Super fast auto-configuring artificial neural networks and their application to power systems. *Electric Power Syst Res*, 35: 11–16, 1995.

[17] L. Cao, Support vector machines experts for time series forecasting. *Neurocomputing*, 51: 321–339, 2003.

[18] L. Cao and Q. Gu, Dynamic support vector machines for non-stationary time series forecasting. Intell Data Anal, 6:67–83, 2002.

[19] F.E.H. Tay, L. Cao, Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48: 847–861, 2002.

[20] F.E.H. Tay and L. Cao, Application of support vector machines in financial time series forecasting. Omega, 29: 309–317, 2001.

[21] W. Lu., W. Wang., AYT. Leung., SM. Lo, RKK. Yuen., Z. Xu and H. Fan., Air pollutant parameter forecasting using support vector machines. *Proceedings of 2002 International Joint Conference*, 1:12–17, 2002.

[22] S.F. Yuan. and FL.Chu, Fault diagnostics based on particle swarm optimization and support vector machines. *Mechanical Systems and Signal Processing,* 21:1787-1798, 2007

[23] Q.B. Zhang, HZ. Cheng, QS. Liu, JW.Zheng and DH.Ni, Short-term load forecasting based on attribute reduction algorithm of rough sets and support vector machine. *Power System Technology*, 8:56-58, 2003.

[24] P.F. Pai and WC. Hong, Support vector machines with simulated annealing algorithms in electricity load forecasting. Taiwan. *Energy Conversion and Management,* 2:2669–2688, 2005.